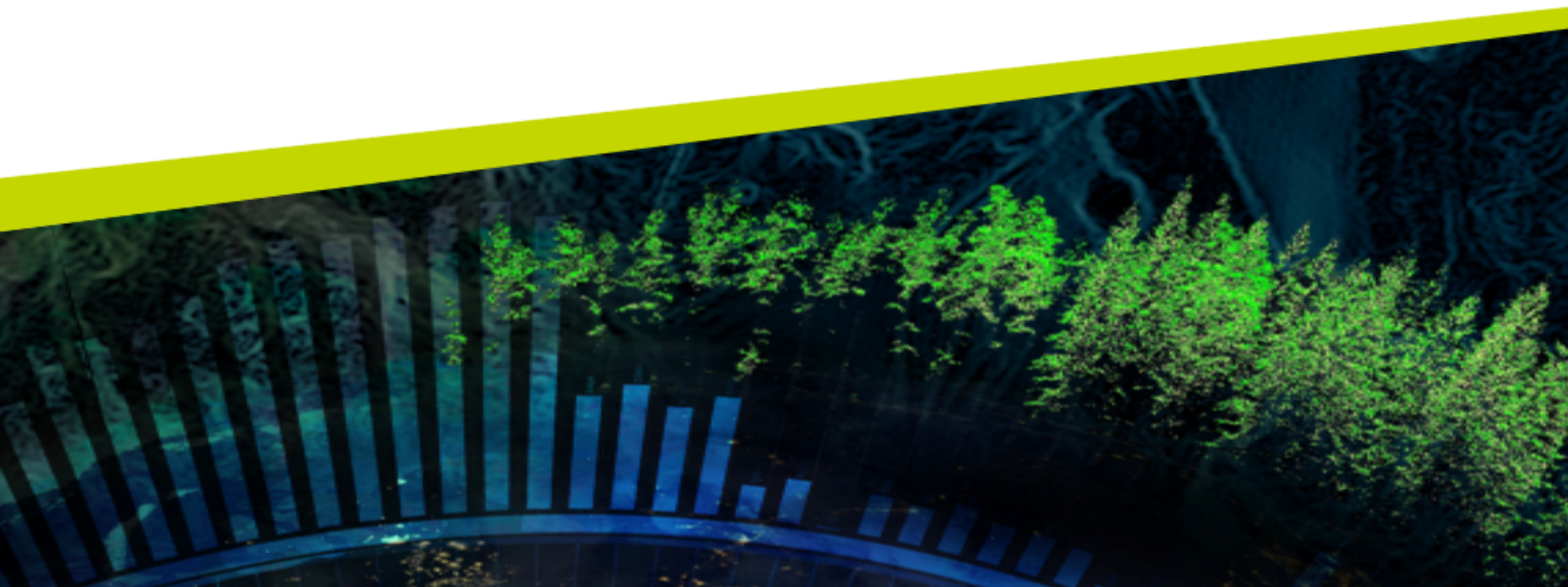
INFERENCIA Y MODELOS ESTADÍSTICOS

Jacqueline Köhler C. y José Luis Jara V.



CAPÍTULO 16. REGRESIÓN LOGÍSTICA

En los capítulos 14 y 15 estudiamos la regresión lineal, útil para predecir una respuesta numérica a partir de una o más variables, aunque con una serie de condiciones. Como explican Field y col. (2012, pp. 312-345), la **regresión logística** es un **modelo lineal generalizado**, que admite una variable de respuesta cuyos residuos sigan una distribución diferente a la normal.

La regresión logística relaciona la distribución de la variable de respuesta con un modelo lineal usando como función de enlace la **función logística estándar**, también conocida como `logit()`, que presentamos en la ecuación 16.1 y mostramos gráficamente en la figura 16.1. Esta función describe una **transición de cero a uno**, por lo que resulta especialmente útil para representar la **probabilidad** de que ocurra algún evento: un valor cercano a cero indica que es muy poco probable, mientras un valor cercano a 1 corresponde a una alta probabilidad (lógicamente, un valor de $0,5$ indica que es igualmente probable que el evento ocurra o no). Así, la regresión logística resulta adecuada para predecir una respuesta dicotómica, pues puede ser asociada a una **distribución binomial**.

$$f(z) = \frac{1}{1 + e^{-z}} \tag{16.1}$$

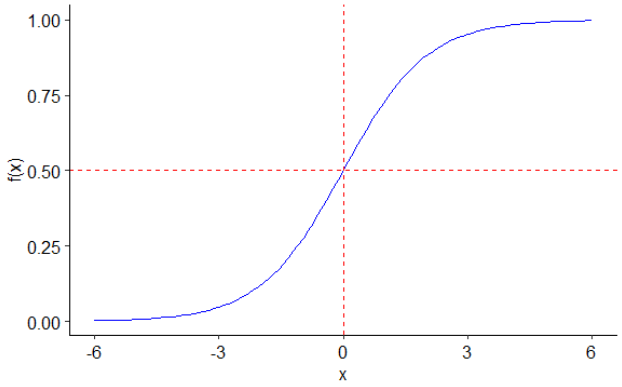


Figura 16.1: función logística.

Para entender mejor esta idea, necesitamos introducir el concepto de ***odds*** (Cerda y col., 2013), el cual no tiene una traducción directa al castellano, pero que puede entenderse como "oportunidad" o "chance", aunque a veces se traduce incorrectamente como "probabilidad". Matemáticamente, el *odds ratio* está dado por la ecuación 16.2, por lo que se define como la razón entre la probabilidad de que ocurra un evento y la probabilidad de que este no ocurra.

$$odds = \frac{p}{1 - p} \tag{16.2}$$

Tomemos el ejemplo que usan (Cerda y col., 2013): supongamos que los registros históricos dicen que en junio llueve 12 días. Así, la probabilidad de que un día de junio sea lluvioso es:

$$p = \frac{12}{30} = 0,4$$

Pero la oportunidad de que el día sea lluvioso es:

$$odds = \frac{12}{18} = 0,67$$

Ambas medidas presentan la misma información, pero de manera diferente. Cuando un evento e tiene las mismas posibilidades de ocurrir o no, $p(e) = 0,5$ y $odds(e) = 1$.

Suponiendo que el logaritmo de los *odds* sigue una distribución normal, podemos relacionar los *odds* y las probabilidades como muestran las ecuaciones 16.3 y 16.4.

$$z = \log(\frac{p}{1-p}) \tag{16.3}$$

$$p = \text{logit}(z) = \frac{1}{1+e^{-z}} \tag{16.4}$$

A su vez, también podemos asociar z a otras variables, como muestra la ecuación 16.5.

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n \tag{16.5}$$

Así, la regresión logística nos permite **asociar** la probabilidad de ocurrencia de un evento e a una combinación lineal de variables predictoras $x_1, x_2, \ldots, x_n$, de acuerdo a la ecuación 16.6.

$$p(e) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\ldots+\beta_n x_n)}} \tag{16.6}$$

De esta forma podemos seleccionar una división o **umbral** que permita **predecir** la ocurrencia de un evento e o, de forma equivalente, **clasificar** un objeto en dos categorías posibles:

- Para valores menores que el umbral se predice que el evento e "no ocurre", otorgándose una clasificación cero ($\hat{y} = 0$) o negativa ($\hat{y} = -$).
- Mientras que para valores mayores o iguales que el umbral se predice que el evento "sí ocurre", a lo que corresponde una clasificación uno ($\hat{y} = 1$) o positiva ($\hat{y} = +$).

Si bien es usual utilizar el valor $p(e) = 0,5$ como umbral, esto no es obligatorio ni siempre conveniente, como veremos más adelante.

En capítulos precedentes explicamos que el ajuste de un modelo de regresión lineal se realiza mediante la resolución de un problema de optimización que busca minimizar la suma de las desviaciones cuadradas entre las respuestas predichas y las observadas. En el caso de la regresión logística, también se ajusta mediante la resolución de un problema de optimización, donde buscamos minimizar la diferencia entre las respuestas observadas y las respuestas predichas. Como las respuestas corresponden a una **variable dicotómica**, esta optimización se realiza usando la función de verosimilitud $\mathcal{L}(p)$, aunque, por conveniencia, se suele optimizar el logaritmo natural de la verosimilitud, tal y como se muestra en la ecuación 16.7.

$$\begin{aligned}
\mathcal{L}(p) &= P(y_1, y_2, \ldots, y_n|p) \text{ with } y_i \in \{0,1\} \\
&= \prod_{i=1}^{n} p^{y_i}(1-p)^{y_i} \\
&\text{Luego,} \\
\ln \mathcal{L}(p) &= \sum_{i=1}^{n} \left[y_i \ln p_{(y_i=1)} + (1-y_i) \ln(1-p_{(y_i=1)}) \right]
\end{aligned} \tag{16.7}$$

16.1 EVALUACIÓN DE UN CLASIFICADOR

Una forma de evaluar modelos de clasificación, entre ellos los de regresión logística, es de acuerdo a la cantidad de errores cometidos (Zelada, 2017). Para ello, el primer paso consiste en construir una tabla de contingencia (también llamada matriz de confusión) para las respuestas predichas y observadas, como muestra la tabla 16.1, bastante similar a la que ya conocimos para explicar los errores de decisión en la prueba de hipótesis (tabla 4.1). Las cuatro celdas de la matriz de confusión contienen:

- **Verdaderos positivos** (VP): cantidad de instancias correctamente clasificadas como pertenecientes a la clase positiva.
- **Falsos positivos** (FP): cantidad de instancias erróneamente clasificadas como pertenecientes a la clase positiva.
- **Falsos negativos** (FN): cantidad de instancias erróneamente clasificadas como pertenecientes a la clase negativa.
- **Verdaderos negativos** (VN): cantidad de instancias correctamente clasificadas como pertenecientes a la clase negativa.

		Real		
		1 (+)	**0 (−)**	**Total**
Clasificación	**1 (+)**	VP	FP	$VP + FP$
	0 (−)	FN	VN	$FN + VN$
	Total	$VP + FN$	$FP + VN$	n

Tabla 16.1: tabla de contingencia para evaluar un clasificador.

La **exactitud** (*accuracy*) del modelo corresponde a la proporción de observaciones correctamente clasificadas, dada por la ecuación 16.8.

$$\text{exactitud} = \frac{VP + VN}{n} \tag{16.8}$$

A su vez, el **error** del modelo corresponde a la proporción de observaciones clasificadas de manera equivocada (ecuación 16.9).

$$\text{error} = \frac{FP + FN}{n} = 1 - \text{exactitud} \tag{16.9}$$

La **sensibilidad** (*sensitivity* o *recall*, ecuación 16.10) indica cuán apto es el modelo para detectar aquellas observaciones pertenecientes a la clase positiva.

$$\text{sensibilidad} = \frac{VP}{VP + FN} \tag{16.10}$$

De manera análoga, la **especificidad** (*specificity*, ecuación 16.11) permite determinar cuán exacta es la asignación de elementos a la clase positiva. También puede entenderse como la aptitud del modelo para correctamente asignar observaciones a la clase negativa.

$$\text{especificidad} = \frac{VN}{FP + VN} \tag{16.11}$$

La **precisión** (*precision*) o valor predictivo positivo (VPP, ecuación 16.12) indica la proporción de instancias clasificadas como positivas que realmente lo son.

$$VPP = \frac{VP}{VP + FP} \tag{16.12}$$

Asimismo, el **valor predictivo negativo** (VPN, ecuación 16.13) señala la proporción de instancias correctamente clasificadas como pertenecientes a la clase negativa.

$$VPN = \frac{VN}{FN + VN} \tag{16.13}$$

Otra herramienta útil es la **curva de calibración**, también llamada curva ROC por las siglas inglesas para *receiver-operating characteristic*, que muestra la relación entre la sensibilidad y la especificidad del modelo (Glen, 2017). Este gráfico también permite evaluar la precisión del modelo, puesto que mientras más se aleje la curva de la diagonal, mayor es la precisión. Para ilustrar mejor la utilidad de este gráfico, la figura 16.2 muestra las curvas ROC para dos modelos diferentes, además de la diagonal. Esta figura indica que el clasificador representado por la curva morada es mejor que el representado por la curva azul, pues se aleja más de la diagonal.

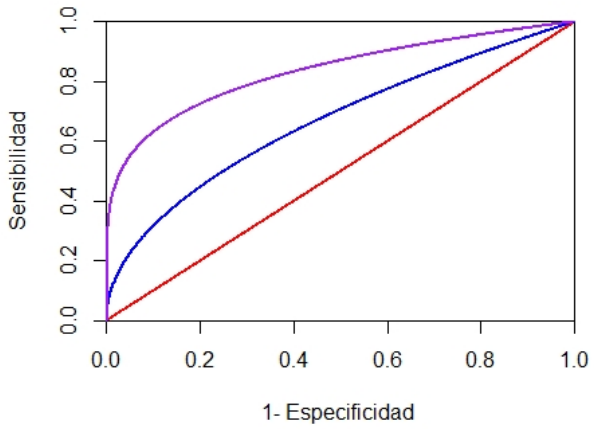


Figura 16.2: dos curvas ROC. Fuente: Ayala (2020).

16.2 BONDAD DE AJUSTE DEL MODELO

Al igual que en el caso de la regresión lineal, existen diversos mecanismos para evaluar el ajuste de un modelo de regresión logística. El estadístico de **log-verosimilitud** ($\ln \mathcal{L}$), dado por la ecuación 16.7, nos permite cuantificar la diferencia entre las probabilidades predichas y las observadas. Este estadístico se asemeja a la suma de los residuos cuadrados de la regresión lineal en el sentido de que cuantifica la cantidad de información que carece de explicación tras el ajuste del modelo. Así, mientras menor sea su valor, mejor es el ajuste del modelo.

La **desviación** (en inglés *deviance*), a menudo denotada por $-2LL$ y en pocas ocasiones llamada *devianza*, suele usarse en lugar de la log-verosimilitud porque sigue una distribución χ^2, lo que facilita calcular el nivel de significación del valor. Está dada por la ecuación 16.14.

$$-2LL = -2 \cdot \ln \mathcal{L} \tag{16.14}$$

Los criterios de evaluación de modelos basados, en el principio de parsimonia que estudiamos en el capítulo 15, también están definidos para la regresión logística. El más sencillo es el criterio de información de Akaike (AIC), dado por la ecuación 16.15, donde k corresponde a la cantidad de predictores en el modelo.

$$AIC = -2LL + 2k \tag{16.15}$$

Similar al AIC, el criterio bayesiano de Schwarz (BIC) ajusta la penalización a la complejidad del modelo según el tamaño de la muestra, como muestra la ecuación 16.16.

$$BIC = -2LL + 2k \cdot \ln n \tag{16.16}$$

16.3 REGRESIÓN LOGÍSTICA EN R

En R, podemos ajustar un modelo de regresión logística mediante la función `glm(formula, family = binomial(link = "logit"), data)`, donde:

- `formula` tiene la forma <variable de respuesta>∼<variable predictora>.
- `data`: matriz de datos.

Puesto que existen otros modelos generalizados de regresión lineal, el argumento `family = binomial(link = "logit")` indica que asumiremos una distribución binomial para la variable de respuesta y que usaremos la función logística.

Las líneas 11–19 del script 16.1 ilustran el uso de la función `glm()` para ajustar un modelo de regresión logística que prediga el tipo de transmisión de un automóvil (0 = automática, 1 = manual) a partir de su peso, usando para ello el ya conocido conjunto de datos `mtcars` disponible en R (recordemos que podemos consultar la descripción de las variables en la tabla 14.1). Para ello, consideramos un conjunto de entrenamiento con 80 % de las instancias. El modelo resultante se muestra en la figura 16.3, donde podemos apreciar que el AIC es bastante bajo ($AIC = 16,23$) y que la desviación del modelo con una variable (23 grados de libertad) es de $12,23$.

Las líneas 22–33 evalúan el modelo ajustado usando las herramientas descritas en la sección anterior y el conjunto de entrenamiento. La función `roc(response, predictor)` del paquete `pROC`, donde los argumentos corresponden, respectivamente, a las respuestas observadas y las respuestas predichas, nos permite obtener la curva ROC de la figura 16.4. La curva se aleja bastante de la diagonal, por lo que al parecer se trata de un buen modelo. A su vez, la función `confusionMatrix(data, reference)` del paquete `caret`, donde `data` corresponde a la respuesta predicha y `reference` a la observada, genera la matriz de confusión y obtiene las medidas de evaluación descritas anteriormente, como muestra la figura 16.5. Podemos ver que el modelo tiene una exactitud de 92,0 %. La sensibilidad de 100 % y la especificidad de 83,33 % muestran que el modelo se desempeña un poco mejor identificando elementos de la clase positiva, correspondiente en este caso a los vehículos de transmisión automática.

Pero, como ya hemos estudiado en capítulos anteriores, debemos evaluar el modelo con un conjunto de datos diferente al que usamos para su construcción. Así, las líneas 36–46 obtienen la curva ROC (figura 16.6) y la matriz de confusión (figura 16.7) para el conjunto de prueba, donde observamos un resultado con menor exactitud que con el conjunto de entrenamiento. Esto es una indicación de que el modelo podría estar un poco sobreajustado para el conjunto de entrenamiento, pero también de que el conjunto de prueba puede ser muy pequeño para obtener una evaluación confiable.

```
Call:
glm(formula = am ~ wt, family = binomial(link = "logit"), data = entrenamiento)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.17498  -0.40172  -0.00176   0.12321   2.26151

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   18.525      8.504   2.178   0.0294 *
wt            -5.883      2.645  -2.224   0.0261 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 34.617  on 24  degrees of freedom
Residual deviance: 12.230  on 23  degrees of freedom
AIC: 16.23

Number of Fisher Scoring iterations: 7
```

Figura 16.3: ajuste de un modelo de regresión logística.

Script 16.1: ajuste de un modelo de regresión logística en R.

```r
library(pROC)
library(caret)

set.seed(1313)

# Cargar los datos.
datos <- mtcars
datos$am <- factor(datos$am)

# Separar conjuntos de entrenamiento y prueba.
n <- nrow(datos)
n_entrenamiento <- floor(0.8 * n)
muestra <- sample.int(n = n, size = n_entrenamiento, replace = FALSE)
entrenamiento <- datos[muestra, ]
prueba  <- datos[-muestra, ]

# Ajustar modelo.
modelo <- glm(am ~ wt, family = binomial(link = "logit"), data = entrenamiento)
print(summary(modelo))

# Evaluar el modelo con el conjunto de entrenamiento.
cat("Evaluación del modelo a partir del conjunto de entrenamiento:\n")
probs_e <- predict(modelo, entrenamiento, type = "response")

umbral <- 0.5
preds_e <- sapply(probs_e, function(p) ifelse(p >= umbral, "1", "0"))
preds_e <- factor(preds_e, levels = levels(datos[["am"]]))

ROC_e <- roc(entrenamiento[["am"]], probs_e)
plot(ROC_e)
```

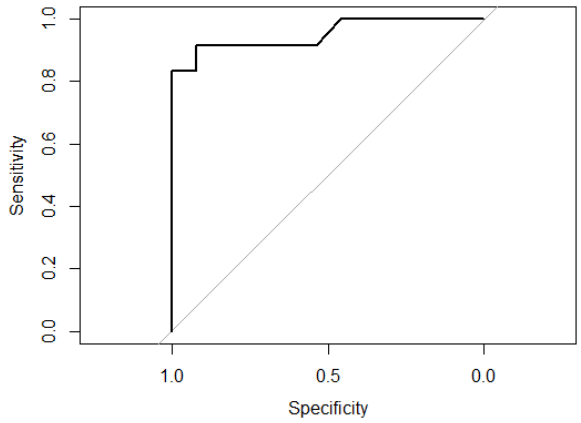


Figura 16.4: curva ROC obtenida al evaluar el modelo con el conjunto de entrenamiento.

```
31
32 matriz_e <- confusionMatrix(preds_e, entrenamiento[["am"]])
33 print(matriz_e)
34
35 # Evaluar el modelo con el conjunto de prueba.
36 cat("Evaluación del modelo a partir del conjunto de prueba:\n")
37 probs_p <- predict(modelo, prueba, type = "response")
38
39 preds_p <- sapply(probs_p, function(p) ifelse(p >= umbral, "1", "0"))
40 preds_p <- factor(preds_p, levels = levels(datos[["am"]]))
41
42 ROC_p <- roc(prueba[["am"]], probs_p)
43 plot(ROC_p)
44
45 matriz_p <- confusionMatrix(preds_p, prueba[["am"]])
46 print(matriz_p)
```

16.4 CONDICIONES PARA USAR REGRESIÓN LOGÍSTICA

Desde luego, no basta con evaluar el desempeño del clasificador, sino que también necesitamos verificar el cumplimiento de ciertas condiciones para que un modelo de regresión logística sea válido:

1. Debe existir una relación lineal entre los predictores y la respuesta transformada.
2. Los residuos deben ser independientes entre sí.

Además de las condiciones anteriores, existen otras situaciones en que puede ocurrir que el método de optimización no converja:

1. Multicolinealidad entre los predictores, que en este caso se aborda del mismo modo que para RLM (por ejemplo, mediante el factor de inflación de la varianza o la tolerancia).
2. Información incompleta, que se produce cuando no contamos con observaciones suficientes para todas las posibles combinaciones de predictores.
3. Separación perfecta, que ocurre cuando no hay superposición entre las clases, es decir, ¡cuando los predictores separan ambas clases completamente!

```
Confusion Matrix and Statistics

          Reference
Prediction  0  1
         0 13  2
         1  0 10

               Accuracy : 0.92
                 95% CI : (0.7397, 0.9902)
    No Information Rate : 0.52
    P-Value [Acc > NIR] : 2.222e-05

                  Kappa : 0.8387

 Mcnemar's Test P-Value : 0.4795

            Sensitivity : 1.0000
            Specificity : 0.8333
         Pos Pred Value : 0.8667
         Neg Pred Value : 1.0000
             Prevalence : 0.5200
         Detection Rate : 0.5200
   Detection Prevalence : 0.6000
      Balanced Accuracy : 0.9167

       'Positive' Class : 0
```

Figura 16.5: matriz de confusión y medidas de evaluación con el conjunto de entrenamiento para el modelo ajustado.

16.5 GENERALIZACIÓN DEL MODELO

En capítulos anteriores conocimos la validación cruzada como herramienta para mejorar la estimación del error, la cual podemos usar de manera análoga para regresión logística. El script 16.2 mejora el ejercicio realizado en el script 16.1, incorporando el uso de validación cruzada de 5 pliegues. Notemos que la llamada a la función `train()` también solicita que "se guarden" los valores predichos, lo que nos permite estimar el rendimiento promedio del modelo como si se repitiera el script 16.2, seleccionando aleatoriamente un conjunto de entrenamiento y otro de prueba, cinco veces.

Debemos fijarnos en que el modelo obtenido es idéntico al anterior (por lo que no se muestra aquí), ya que la función `train()` reentrena el modelo del pliegue que obtuvo mejor rendimiento con todos los datos disponibles. En el caso de la regresión logística (como con la regresión lineal), los pliegues solo se diferencian en los datos que utilizan, por lo que siempre se llega al mismo modelo. Esto no sería así si la validación cruzada se usara, por ejemplo, para seleccionar las variables predictoras a incluir en el modelo.

Script 16.2: ajuste de un modelo de regresión logística usando validación cruzada.

```
1 library(caret)
2
3 set.seed(1313)
4
5 # Cargar los datos.
6 datos <- mtcars
7 datos$am <- factor(datos$am)
```

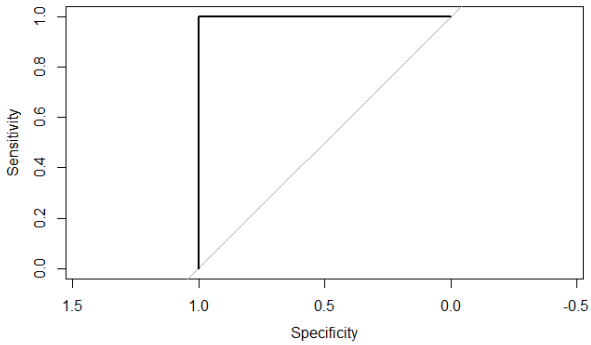


Figura 16.6: curva ROC obtenida al evaluar el modelo con el conjunto de prueba.

```
8
9  # Ajustar modelo usando validación cruzada de 5 pliegues.
10 modelo <- train(am ~ wt, data = entrenamiento, method = "glm",
11                  family = binomial(link = "logit"),
12                  trControl = trainControl(method = "cv", number = 5,
13                                           savePredictions = TRUE))
14
15 print(summary(modelo))
16
17 # Evaluar el modelo
18 cat("Evaluación del modelo basada en validación cruzada:\n")
19 matriz <- confusionMatrix(modelo$pred$pred, modelo$pred$obs)
20 print(matriz)
```

16.6 SELECCIÓN DE PREDICTORES

Cuando tenemos múltiples predictores potenciales, debemos decidir cuáles de ellos incorporar en el modelo. Una vez más, y tal como detallamos en el capítulo 15, el ideal es usar la regresión jerárquica para escoger los predictores de acuerdo a evidencia disponible en la literatura. Sin embargo, al explorar los datos, podemos emplear los demás métodos ya descritos: selección hacia adelante, eliminación hacia atrás, regresión escalonada o todos los subconjuntos. Se usan para ello las mismas funciones de R descritas en el capítulo 15.

16.7 COMPARACIÓN DE MODELOS

Al igual que con los modelos de regresión lineal, podemos comparar modelos de regresión logística mediante la función `anova()`, aunque ahora la prueba F resulta inapropiada. En cambio, una prueba muy utilizada en este caso es el *Likelihood Ratio Test* (LRT), el cual compara qué tanto más "probables" son los datos con un modelo que con el otro. Podemos ver un ejemplo de esta comparación más adelante en el script 16.3.

```
Confusion Matrix and Statistics

          Reference
Prediction 0 1
         0 5 0
         1 1 1

               Accuracy : 0.8571
                 95% CI : (0.4213, 0.9964)
    No Information Rate : 0.8571
    P-Value [Acc > NIR] : 0.7365

                  Kappa : 0.5882

 Mcnemar's Test P-Value : 1.0000

            Sensitivity : 0.8333
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 0.5000
             Prevalence : 0.8571
         Detection Rate : 0.7143
   Detection Prevalence : 0.7143
      Balanced Accuracy : 0.9167

       'Positive' Class : 0
```

Figura 16.7: matriz de confusión y medidas de evaluación con el conjunto de prueba para el modelo ajustado.

16.8 REGRESIÓN LOGÍSTICA EN R CON SELECCIÓN DE PREDICTORES

En páginas previas ajustamos un modelo de regresión logística para determinar el tipo de transmisión de un automóvil a partir de su peso. Sin embargo, el predictor fue seleccionado de manera aleatoria, simplemente para ilustrar el proceso, por lo que podríamos encontrar un mejor modelo usando algún método de selección de predictores. Las líneas 19–32 del script 16.3 llevan a cabo esta tarea usando regresión escalonada, obteniéndose como resultado el modelo presentado en la figura 16.8.

Sin embargo, al ajustar este modelo R emite algunas advertencias, como muestra la figura 16.9. Estas ocurren cuando los predictores separan completamente las clases, o bien cuando existen problemas de colinealidad. Al verificar los factores de inflación de la varianza de los predictores (script 16.3, líneas 35–41) podemos apreciar que, si bien ninguna de las variables presenta un *VIF* superior a 10, el promedio es bastante superior a 1 (figura 16.10), lo que confirma que el modelo puede tener problemas. En consecuencia, no es recomendable usar este modelo.

Por regla general, se recomienda eliminar la variable con mayor *VIF*, pero en este caso ambos son iguales. En consecuencia, en las líneas 44–57 del script 16.3 se ajustan los dos modelos posibles (figuras 16.11 y 16.12) y luego se comparan (figura 16.13). Pero en este caso la prueba ANOVA no sirve, pues ambos modelos tienen igual cantidad de predictores y no entrega un valor p. Sin embargo, podemos ver que el *VIF* del modelo con la potencia como predictor ($VIF = 37,444$) es más alto que para el modelo con el peso como predictor ($VIF = 16,23$). En consecuencia, este último parece ser mejor.

A modo de ejercicio, a pesar de que lo descartamos por tener problemas de colinealidad, comparamos el

```
Call:
glm(formula = am ~ wt + hp, family = binomial(link = "logit"),
    data = entrenamiento)

Deviance Residuals:
      Min          1Q      Median          3Q         Max
-3.378e-05  -2.100e-08  -2.100e-08   2.100e-08   2.597e-05

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  4.165e+02  3.478e+05   0.001    0.999
wt          -1.555e+02  1.287e+05  -0.001    0.999
hp           4.788e-01  4.620e+02   0.001    0.999

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 3.4617e+01  on 24  degrees of freedom
Residual deviance: 2.2982e-09  on 22  degrees of freedom
AIC: 6

Number of Fisher Scoring iterations: 25
```

Figura 16.8: modelo de regresión logística obtenido mediante regresión escalonada.

```
Warning messages:
1: glm.fit: algorithm did not converge
2: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

Figura 16.9: modelo de regresión logística obtenido mediante regresión escalonada.

modelo obtenido mediante regresión escalonada con el que tiene al peso como variable predictora (script 16.3, línea 68), obteniendo como resultado un valor $p < 0,001$ (figura 16.14). Puesto que el valor p obtenido es significativo, la prueba arroja que el modelo más complejo (es decir, el que tiene dos predictores) reduce la varianza de los residuos de forma significativa (¡llegando a cero!).

Dado que el mejor modelo es el mismo que habíamos usado en secciones previas, no repetiremos la evaluación con el conjunto de prueba, pues ya presentamos el resultado en la figura 16.7.

Sin embargo, aún resta verificar el cumplimiento de la condición de independencia de los residuos, para lo cual, al igual que con modelos de regresión lineal, empleamos la prueba de Durbin-Watson (script 16.3, línea 75), cuyo resultado mostramos en la figura 16.15, donde podemos notar que, aunque cerca del borde para $\alpha = 0,05$, los residuos son independientes.

```
Verificación de colinealidad
-------------------------------------

VIF:
     wt      hp
4.14191 4.14191

Promedio VIF: [1] 4.14191
```

Figura 16.10: factores de inflación de la varianza para los modelos.

```
Call:
glm(formula = am ~ wt, family = binomial(link = "logit"), data = entrenamiento)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.17498  -0.40172  -0.00176   0.12321   2.26151

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   18.525      8.504   2.178   0.0294 *
wt            -5.883      2.645  -2.224   0.0261 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 34.617  on 24  degrees of freedom
Residual deviance: 12.230  on 23  degrees of freedom
AIC: 16.23

Number of Fisher Scoring iterations: 7
```

Figura 16.11: modelo de regresión logística con el peso como predictor.

Una vez verificadas las condiciones, podemos concluir que el modelo es adecuado y puede ser generalizado. No obstante, aún falta determinar si su ajuste se ve afectado por la presencia de valores atípicos (script 16.3, líneas 78–137). La figura 16.16 muestra los gráficos asociados al modelo. El gráfico de la figura 16.16a muestra una única instancia cuyo residuo se aleja muchísimo de los demás, correspondiente al Maserati Bora, el cual también se aleja bastante de la recta esperada en el gráfico Q-Q de los residuos (figura 16.16b). A su vez, la figura 16.16d muestra claramente que esa misma instancia ejerce un importante apalancamiento.

Usando herramientas más precisas para replicar el análisis, parte de cuyos resultados presentamos en la figura 16.17[1], podemos determinar que la única observación cuyo residuo estandarizado escapa a la normalidad es el Maserati Bora. Asimismo, esta instancia presenta la mayor distancia de Cook y los mayores DFBeta, por lo que es la única que resulta preocupante y podría ser útil eliminarla para el ajuste del modelo. También queda como ejercicio reentrenar y evaluar el modelo sin considerar esta observación.

Script 16.3: ajuste y evaluación del mejor modelo para predecir el tipo de transmisión de un automóvil.

```r
library(car)

set.seed(1313)

# Cargar los datos.
datos <- mtcars
am <- factor(datos$am)
datos$am <- NULL
datos <- cbind(am, datos)

# Separar conjuntos de entrenamiento y prueba.
n <- nrow(datos)
n_entrenamiento <- floor(0.8 * n)
muestra <- sample.int(n = n, size = n_entrenamiento, replace = FALSE)
entrenamiento <- datos[muestra, ]
```

[1] Por una cuestión de espacio, queda como ejercicio para el lector ejecutar el script 16.3 y ver el detalle de los valores obtenidos para las distintas métricas evaluadas para las observaciones sospechosas.

```
Call:
glm(formula = am ~ hp, family = binomial(link = "logit"), data = entrenamiento)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3663  -1.0648  -0.8953   1.1182   1.7415

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.820490   0.941997   0.871    0.384
hp          -0.006236   0.005965  -1.045    0.296

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 34.617  on 24  degrees of freedom
Residual deviance: 33.444  on 23  degrees of freedom
AIC: 37.444

Number of Fisher Scoring iterations: 4
```

Figura 16.12: modelo de regresión logística con la potencia como predictor.

```
Analysis of Deviance Table

Model 1: am ~ wt
Model 2: am ~ hp
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        23     12.230
2        23     33.444  0  -21.214
```

Figura 16.13: comparación de los modelos con un único predictor.

```
prueba  <- datos[-muestra, ]

# Ajustar modelo nulo.
nulo <- glm(am ~ 1, family = binomial(link = "logit"), data = entrenamiento)

# Ajustar modelo completo.
cat("\n\n")
completo <- glm(am ~ ., family = binomial(link = "logit"),
                data = entrenamiento)

# Ajustar modelo con regresión escalonada.
cat("Modelo con regresión escalonada\n")
cat("-------------------------------------\n")
mejor <- step(nulo, scope = list(lower = nulo, upper = completo),
              direction = "both", trace = 0)

print(summary(mejor))

# Verificación de multicolinealidad.
cat("Verificación de colinealidad\n")
cat("-------------------------------------\n")
cat("\nVIF:\n")
vifs <- vif(mejor)
```

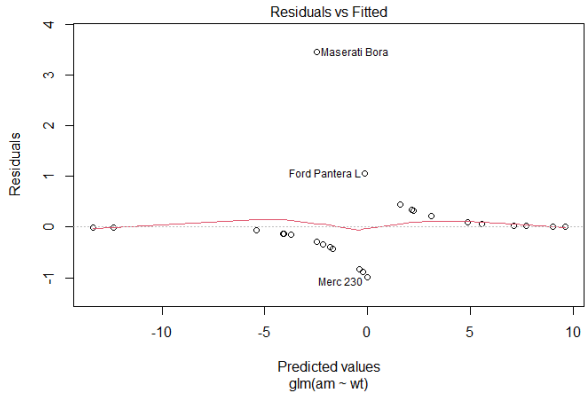
```
Analysis of Deviance Table

Model 1: am ~ wt
Model 2: am ~ wt + hp
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1        23      12.23
2        22       0.00  1    12.23 0.0004704 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figura 16.14: comparación del modelo con dos predictores y el que solo tiene el peso como predictor.

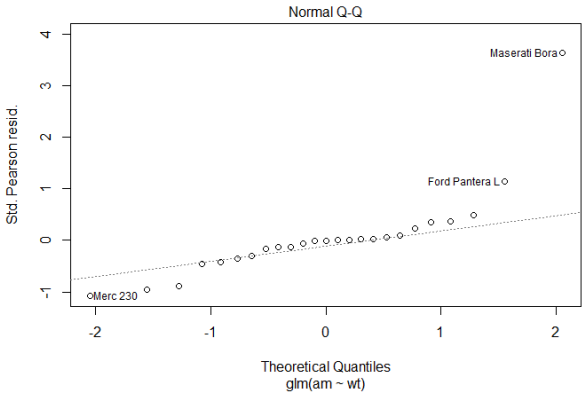
```
 lag Autocorrelation D-W Statistic p-value
   1     -0.30715746      2.583329   0.084
 Alternative hypothesis: rho[lag] != 0
```

Figura 16.15: resultado de la prueba de Durbin-Watson para verificar la independencia de los residuos del modelo que solo tiene el peso como predictor.

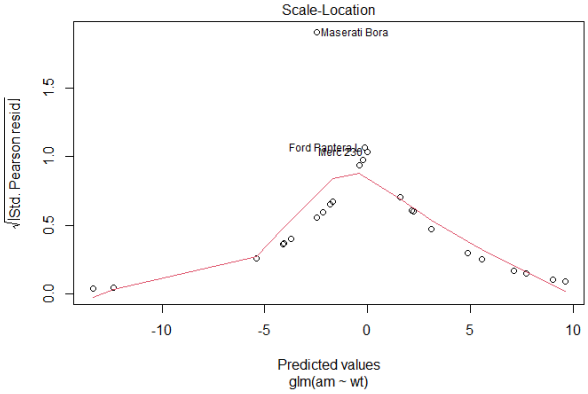
```
print(vifs)
cat("\nPromedio VIF: ")
print(mean(vifs))

# Ajustar modelo con el peso como predictor.
cat("Modelo con el peso como predictor\n")
cat("-----------------------------------\n")
modelo_peso <- glm(am ~ wt, family = binomial(link = "logit"),
                   data = entrenamiento)

print(summary(modelo_peso))

# Ajustar modelo con la potencia como predictor.
cat("Modelo con la potencia como predictor\n")
cat("-------------------------------------\n")
modelo_potencia <- glm(am ~ hp, family = binomial(link = "logit"),
                       data = entrenamiento)

print(summary(modelo_potencia))

# Comparar los modelos con el peso y la potencia como predictores.
cat("\n\n")
cat("Likelihood Ratio Test para los modelos\n")
cat("--------------------------------------\n")
print(anova(modelo_peso, modelo_potencia, test = "LRT"))

# A modo de ejercicio, comparar el modelo obtenido mediante
# regresión escalonada con el que solo tiene el peso como predictor.
cat("\n\n")
cat("Likelihood Ratio Test para los modelos\n")
cat("--------------------------------------\n")
print(anova(modelo_peso, mejor, test = "LRT"))

# Independencia de los residuos.
cat("Verificación de independencia de los residuos\n")
cat("---------------------------------------------\n")
print(durbinWatsonTest(modelo_peso, max.lag = 5))
```



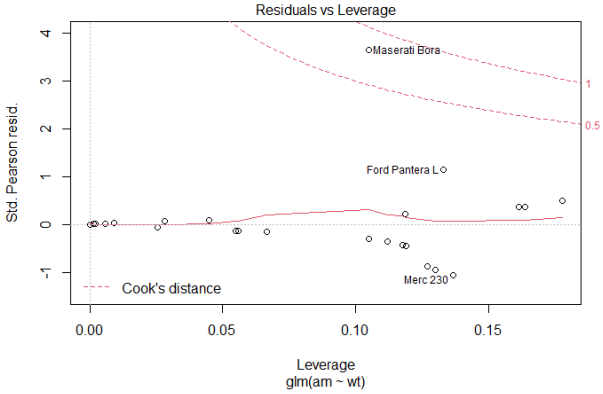
(a) residuos.



(b) distribución de los residuos.



(c) residuos estandarizados.



(d) apalancamiento.

Figura 16.16: gráficos para evaluar el modelo de regresión logística.

```
76
77 # Detectar posibles valores atípicos.
78 cat("Identificación de posibles valores atípicos\n")
79 cat("-------------------------------------\n")
80 plot(mejor)
81
82 # Obtener los residuos y las estadísticas.
83 output <- data.frame(predicted.probabilities = fitted(modelo_peso))
84 output[["standardized.residuals"]] <- rstandard(modelo_peso)
85 output[["studentized.residuals"]] <- rstudent(modelo_peso)
86 output[["cooks.distance"]] <- cooks.distance(modelo_peso)
87 output[["dfbeta"]] <- dfbeta(modelo_peso)
88 output[["dffit"]] <- dffits(modelo_peso)
89 output[["leverage"]] <- hatvalues(modelo_peso)
90
91 # Evaluar residuos estandarizados que escapen a la normalidad.
92 # 95% de los residuos estandarizados deberían estar entre
93 # -1.96 y 1.96, y 99% entre -2.58 y 2.58.
94 sospechosos1 <- which(abs(output[["standardized.residuals"]]) > 1.96)
95 sospechosos1 <- sort(sospechosos1)
96 cat("\n\n")
97 cat("Residuos estandarizados fuera del 95% esperado\n")
```

```
Residuales estandarizados fuera del 95% esperado
------------------------------------------------
[1] "Maserati Bora"

Residuales con una distancia de Cook alta
-----------------------------------------
character(0)

Residuales con levarage fuera de rango (> 0.344)
------------------------------------
character(0)

Residuales con DFBeta sobre 1
-----------------------------
[1] "Dodge Challenger" "Maserati Bora"    "Merc 280"
[4] "Valiant"          "Ferrari Dino"     "Volvo 142E"
[7] "Mazda RX4 Wag"

Casos sospechosos
-----------------
                  am  mpg cyl  disp  hp drat    wt  qsec vs gear carb
Dodge Challenger  0 15.5   8 318.0 150 2.76 3.520 16.87  0    3    2
Maserati Bora     1 15.0   8 301.0 335 3.54 3.570 14.60  0    5    8
Merc 280          0 19.2   6 167.6 123 3.92 3.440 18.30  1    4    4
Valiant           0 18.1   6 225.0 105 2.76 3.460 20.22  1    3    1
Ferrari Dino      1 19.7   6 145.0 175 3.62 2.770 15.50  0    5    6
Volvo 142E        1 21.4   4 121.0 109 4.11 2.780 18.60  1    4    2
Mazda RX4 Wag     1 21.0   6 160.0 110 3.90 2.875 17.02  0    4    4
```

Figura 16.17: identificación de posibles valores atípicos.

```
98  cat("-------------------------------------------------\n")
99  print(rownames(entrenamiento[sospechosos1, ]))
100
101 # Revisar casos con distancia de Cook mayor a uno.
102 sospechosos2 <- which(output[["cooks.distance"]] > 1)
103 sospechosos2 <- sort(sospechosos2)
104 cat("\n\n")
105 cat("Residuales con una distancia de Cook alta\n")
106 cat("-----------------------------------------\n")
107 print(rownames(entrenamiento[sospechosos2, ]))
108
109 # Revisar casos cuyo apalancamiento sea más del doble
110 # o triple del apalancamiento promedio.
111 leverage.promedio <- ncol(entrenamiento) / nrow(datos)
112 sospechosos3 <- which(output[["leverage"]] > leverage.promedio)
113 sospechosos3 <- sort(sospechosos3)
114 cat("\n\n")
115 cat("Residuales con levarage fuera de rango (> ")
116 cat(round(leverage.promedio, 3), ")", "\n", sep = "")
117 cat("------------------------------------\n")
118 print(rownames(entrenamiento[sospechosos3, ]))
119
120 # Revisar casos con DFBeta >= 1.
121 sospechosos4 <- which(apply(output[["dfbeta"]] >= 1, 1, any))
```

```r
sospechosos4 <- sort(sospechosos4)
names(sospechosos4) <- NULL
cat("\n\n")
cat("Residuales con DFBeta sobre 1\n")
cat("-----------------------------\n")
print(rownames(entrenamiento[sospechosos4, ]))

# Detalle de las observaciones posiblemente atípicas.
sospechosos <- c(sospechosos1, sospechosos2, sospechosos3, sospechosos4)
sospechosos <- sort(unique(sospechosos))
cat("\n\n")
cat("Casos sospechosos\n")
cat("-----------------\n")
print(entrenamiento[sospechosos, ])
cat("\n\n")
print(output[sospechosos, ])
```

16.9 EJERCICIOS PROPUESTOS

1. ¿Qué es un modelo lineal generalizado?
2. ¿Qué modela una regresión logística?
3. Explica por qué este modelo lleva el apellido "logística".
4. ¿Por qué se considera un modelo lineal?
5. Menciona las condiciones necesarias para aplicar regresión logística.
6. Explica las evaluaciones que deben aplicarse a un modelo de regresión logística.
7. Investiga cuáles son las hipótesis que se contrastan al hacer inferencia con la regresión logística.
8. ¿Se podría buscar un buen modelo de regresión logística usando el paquete `caret`? Investigue.
9. Reconstruye el último modelo de regresión logística que conseguimos, pero ahora sin considerar el "Maserati Bora" y evalúa si cumple las condiciones para ser usado.

REFERENCIAS

Ayala, J. (2020). *Minería de datos*.
Consultado el 23 de junio de 2021, desde https://rpubs.com/JairoAyala/592802

Cerda, J., Vera, C. & Rada, G. (2013). Odds ratio: aspectos teóricos y prácticos.
Revista médica de Chile, *141*, 1329-1335.

Field, A., Miles, J. & Field, Z. (2012). *Discovering statistics using R*. SAGE Publications Ltd.

Glen, S. (2017). *Receiver Operating Characteristic (ROC) Curve: Definition, Example*. Consultado el 23 de junio de 2021, desde https://www.statisticshowto.com/receiver-operating-characteristic-roc-curve/

Zelada, C. (2017). *Evaluación de modelos de clasificación*.
Consultado el 23 de junio de 2021, desde https://rpubs.com/chzelada/275494